


## Chapter 38

# Impact Analysis of Intelligent Agents in Automatic Fault-Prone Components Prediction and Testing: Impact Analysis of Intelligent Agents in Test Automation

**Jeya Mala Dharmalingam**

 <https://orcid.org/0000-0002-2100-8218>

*Thiagarajar College of Engineering, India*

### **ABSTRACT**

*Software quality is imperative for industrial strength software. This quality will be often determined by a few components present in the software which decides the entire functionality. If any of these components are not rigorously tested, the quality will be highly affected. Without knowing which of these components are really critical, it will not be possible to perform high level testing. Hence, to predict such fault-prone or critical components from the software prior to testing and prioritizing them during the testing process, an agent-based approach is proposed in this chapter. The framework developed as part of this work will certainly reduce the field failures and thus will improve the software quality. Further, this approach has also utilized important metrics to predict such components and also prioritized the components based on their critical value. Also, the work proposed in this research has also been compared with some of the existing approaches and the results reveal that, this work is a novel one and can both predict and test the components from the software.*

DOI: 10.4018/978-1-6684-3702-5.ch038

## INTRODUCTION

As per the study of National Institute of Standards and Technology, the cost for an inadequate infrastructure for software testing is estimated to be from \$22.2 to \$59.5 billion (Tassey, 2002). As exhaustive testing (testing 100%) is not feasible (David C., Jinlin Yang, Sarfraz Khurshid, Wei Le and Kevin Sullivan, 2005; Myers, 1979), the industries are forced to stop the testing process at one point of time and deliver the software to the customers. This leads them to compromise the quality of the software due to customers' need for quick delivery of quality software, reduced software development lifecycle, changing markets with global competition and rapid development of new processes and technologies.

The surveys have indicated that, many of the complex systems' failures are due to insufficient testing of software before they are deployed to the customer side (Bernardi, 2011; Schneidewind, 1978). After the analysis, it has been identified that, the highly critical components are not being properly tested or simply ignored without knowing their critical level due to time and cost compromises.

The identification of such critical components from the software prior to testing is still a research area, since the automated testing tools available in the market doesn't address the said problem. Based on our field surveys conducted in several organizations during the past months, it has been identified that, most of the defects reported by the customers after delivery are present in these higher critical components. This gives us the insight on the importance of critical components identification and their verification prior to the delivery of the software. But, the identification of the criticality level of a component involves the evaluation of various metrics and measures associated with them.

Hence, an automated software testing framework that can identify and prioritize the critical components based on the various metrics and measures associated with each of the components and can also provide an optimized critical paths list which can reduce the time and cost needed in the testing process without compromising the testing of these critical components is the need of the hour.

From the literature survey, it has been identified that only a very few works have been conducted in the said research area, and that too have been limited by the type and number of metrics used by them. If the software under test is small and simple then the identification can be done manually. As the real time complex systems have huge functionalities, there is a need for an approach that embeds both intelligence and automation as a tool.

Since software testing is NP-hard (Non-Polynomial hard) (Nagappan, 2006), and as manual testing is costly and error prone, several existing research works on structural testing have employed computationally intelligent techniques, such as artificial intelligence and evolutionary computation methods, to achieve optimization in the testing process (Alok Singh, 2009; Basturk & Karaboga, 2006; Baykasolu A, Lale Özbakır & Pınar Tapkan, 2009; Dorigo M., Maniezzo, V., Colorni, A., 1996, Teodorović & Dell, 2006; Fathian M, Babak Amiri & Ali Maroosi., 2007; Karaboga, 2008, 2009; Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., & Zaidi M., 2006; R. Srinivasa Rao, S.V.L. Narasimham, M. Ramalingaraju., 2008; Wong Li-Pei, Chi Yung Puan, Malcolm Yoke Hean Low, Chin Soon Chong., 2008).

As several existing knowledge based approaches are population based approaches with sequential execution behavior, from the literature survey it has been identified that, agents based approach gives promising results when compared to the other solutions. However, this research work applied the multi agents based approach in which each agent is responsible for doing its activities in a parallel manner.

Hence, this research work proposed, an automated critical components identification and verification framework using multi-agents based approach to identify the critical components and to generate and optimize the number of critical paths needed for their verification.

29 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/impact-analysis-of-intelligent-agents-in-automatic-fault-prone-components-prediction-and-testing/294493](http://www.igi-global.com/chapter/impact-analysis-of-intelligent-agents-in-automatic-fault-prone-components-prediction-and-testing/294493)

## Related Content

---

### Migration Goals and Risk Management in Cloud Computing: A Review of State of the Art and Survey Results on Practitioners

Shareeful Islam, Stefan Fenz, Edgar Weippland Christos Kalloniatis (2016). *International Journal of Secure Software Engineering* (pp. 44-73).

[www.irma-international.org/article/migration-goals-and-risk-management-in-cloud-computing/160712](http://www.irma-international.org/article/migration-goals-and-risk-management-in-cloud-computing/160712)

### Engineering Reusable Learning Objects

Ed Morris (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 718-735).

[www.irma-international.org/chapter/engineering-reusable-learning-objects/29418](http://www.irma-international.org/chapter/engineering-reusable-learning-objects/29418)

### The Recovery Language Approach

Vincenzo De Florio (2009). *Application-Layer Fault-Tolerance Protocols* (pp. 175-241).

[www.irma-international.org/chapter/recovery-language-approach/5126](http://www.irma-international.org/chapter/recovery-language-approach/5126)

### Integrating Access Control into UML for Secure Software Modeling and Analysis

Thuong Doan, Steven Demurjian, Laurent Micheland Solomon Berhe (2010). *International Journal of Secure Software Engineering* (pp. 1-19).

[www.irma-international.org/article/integrating-access-control-into-uml/39006](http://www.irma-international.org/article/integrating-access-control-into-uml/39006)

### Efficient Scheduling of Jobs and Allocation of Resources in Cloud Computing

Sandeep Gajanan Sutarand Kumarswamy S. (2022). *International Journal of Software Innovation* (pp. 1-13).

[www.irma-international.org/article/efficient-scheduling-of-jobs-and-allocation-of-resources-in-cloud-computing/307013](http://www.irma-international.org/article/efficient-scheduling-of-jobs-and-allocation-of-resources-in-cloud-computing/307013)